

WEST[Help](#)[Logout](#)[Interrupt](#)[Main Menu](#)[Search Form](#)[Posting Counts](#)[Show S Numbers](#)[Edit S Numbers](#)[Preferences](#)[Cases](#)**Search Results -**

Terms	Documents
L6 AND testing	20

Database:

US Patents Full-Text Database	▲
US Pre-Grant Publication Full-Text Database	
JPO Abstracts Database	
EPO Abstracts Database	
Derwent World Patents Index	
IBM Technical Disclosure Bulletins	▼

Search:

L7	▲
	▼

[Refine Search](#)[Recall Text](#)[Clear](#)**Search History**
DATE: Sunday, December 14, 2003 [Printable Copy](#) [Create Case](#)
Set Name Query

side by side

*DB=USPT; PLUR=NO; OP=OR*L7 L6 AND testingL6 L5 AND visualL5

(5233686 5638523 5841434 5883639 5485600 5495567 5768510
 5848246 5983245 5987245 5999972 6038590 6052711 6189018
 6233620 6249283 6253282 6266709 6272555 6272556 6278455
 6304893 6424991 6434598 4870561 5862379 6059838 6059838
 4315315 5587935 5991535 5999174 6047124 6189138 5353401
 5566248 5600778 5603034 5644334 5666501 5802334 6088028
 6094197 6100887 6118446 6125375 6231569 6243076 5224039
 5379432).pn.

L4 L3 AND L1L3 ((717/100 |717/101 |717/102 |717/103)!.CCLS.)L2 L1 AND testingL1 ((717/113)!.CCLS.)**Hit Count Set Name**

result set

20 L735 L649 L53 L4235 L325 L272 L1

END OF SEARCH HISTORY

WEST[Help](#)[Logout](#)[Interrupt](#)[Main Menu](#)[Search Form](#)[Posting Counts](#)[Show S Numbers](#)[Edit S Numbers](#)[Preferences](#)[Cases](#)**Search Results -**


Terms	Documents
L3 AND L1	3

Database:

US Patents Full-Text Database	▲
US Pre-Grant Publication Full-Text Database	
JPO Abstracts Database	
EPO Abstracts Database	
Derwent World Patents Index	
IBM Technical Disclosure Bulletins	▼

Search:

L4	▲
	▼

[Refine Search](#)[Recall Text](#) [Clear](#)**Search History****DATE:** Sunday, December 14, 2003 [Printable Copy](#) [Create Case](#)**Set Name Query**

side by side

DB=USPT; PLUR=NO; OP=OR

L4 L3 AND L1L3 ((717/100 |717/101 |717/102 |717/103)!.CCLS.)L2 L1 AND testingL1 ((717/113)!.CCLS.)**Hit Count Set Name**

result set

3 L4235 L325 L272 L1

END OF SEARCH HISTORY

before PLUS

WEST[Help](#)[Logout](#)[Interrupt](#)[Main Menu](#)[Search Form](#)[Posting Counts](#)[Show S Numbers](#)[Edit S Numbers](#)[Preferences](#)[Cases](#)**Search Results -**

Terms	Documents
L1 AND testing	25

Database:

US Patents Full-Text Database
US Pre-Grant Publication Full-Text Database
JPO Abstracts Database
EPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search:

L2

[Refine Search](#)[Recall Text](#)[Clear](#)**Search History****DATE:** Sunday, December 14, 2003 [Printable Copy](#) [Create Case](#)**Set Name Query**
side by side**Hit Count Set Name**
result set*DB=USPT; PLUR=NO; OP=OR*

<u>L2</u>	L1 AND testing	25	<u>L2</u>
<u>L1</u>	((717/113)!.CCLS.)	72	<u>L1</u>

END OF SEARCH HISTORY

Generate Collection

Print

Search Results - Record(s) 1 through 25 of 25 returned.

☐ 1. Document ID: US 6658645 B1

L2: Entry 1 of 25

File: USPT

Dec 2, 2003

US-PAT-NO: 6658645

DOCUMENT-IDENTIFIER: US 6658645 B1

TITLE: METHOD FOR AUTOMATICALLY GENERATING CORRECTED PROGRAM INHERITED PROGRAM,
AUTOMATIC PROGRAM GENERATION APPARATUS, AND STORAGE MEDIUM STORING PROGRAM FOR
AUTOMATICALLY GENERATING CORRECTED PROGRAM INHERITED PROGRAM

DATE-ISSUED: December 2, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Akuta; Hiromi	Tokyo			JP
Izumi; Masaki	Tokyo			JP

US-CL-CURRENT: 717/106; 714/25, 714/38, 714/48, 717/109, 717/110, 717/113, 717/124,
717/125, 717/126

ABSTRACT:

An automatic program generation unit automatically generates a program according to designing information. When a code is added by a change in designing information, a corrected portion inheriting unit retrieves an added portion from a new program automatically generated according to the latest designing information. When a code is deleted by a change in designing information, the deleted portion is retrieved from a corrected program obtained by correcting the previous automatically-generated program. Then, the added portion obtained by the retrieval is added to the corrected program, and the deleted portion is deleted from the corrected program.

11 Claims, 56 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 53

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	WMC	Drawing Deso	Image
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	-----	--------------	-------

☐ 2. Document ID: US 6618853 B1

L2: Entry 2 of 25

File: USPT

Sep 9, 2003

US-PAT-NO: 6618853

DOCUMENT-IDENTIFIER: US 6618853 B1

TITLE: Program production system for semiconductor tester

DATE-ISSUED: September 9, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Ohyama; Chiaki	Tokyo			JP
Hasebe; Junichi	Tokyo			JP

US-CL-CURRENT: 717/109; 700/121, 717/113, 717/125, 717/129

ABSTRACT:

There is disclosed a program production system for a semiconductor tester which performs production of programs while various kinds of information are viewed on screens. A user sets an execution sequence for commands on a target by specifying any from among the program cells included on a sequence setting screen and sets the functions used in semiconductor testing corresponding to those commands and the parameters accompanying the functions by using a function setting screen.

5 Claims, 13 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 13

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWMC	Draw Desc	Image
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	------	-----------	-------

☐ 3. Document ID: US 6550057 B1

L2: Entry 3 of 25

File: USPT

Apr 15, 2003

US-PAT-NO: 6550057
DOCUMENT-IDENTIFIER: US 6550057 B1

TITLE: Piecemeal retrieval in an information services patterns environment

DATE-ISSUED: April 15, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bowman-Amuah; Michel K.	Colorado Springs	CO		

US-CL-CURRENT: 717/126; 700/80, 707/5, 717/101, 717/102, 717/108, 717/109, 717/113

ABSTRACT:

A system, method and article of manufacture are provided for providing a warning upon retrieval of objects that are incomplete. An object is provided with at least one missing attribute. Upon receipt of a request from an application for the object access to the attributes of the object is allowed by the application. A warning is provided upon an attempt to access the attribute of the object that is missing.

15 Claims, 195 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 123

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

KWMC	Draw Desc	Image
------	-----------	-------

☐ 4. Document ID: US 6430556 B1

L2: Entry 4 of 25

File: USPT

Aug 6, 2003

US-PAT-NO: 6430556
DOCUMENT-IDENTIFIER: US 6430556 B1

TITLE: System and method for providing a query object development environment

DATE-ISSUED: August 6, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Goldberg; Robert N.	Redwood City	CA		
Lam; Gloria Y.	Milipitas	CA		

US-CL-CURRENT: 707/4; 707/102, 717/108, 717/109, 717/113, 717/124

ABSTRACT:

A query object generator tool which generates interface definitions and code that implement a query object also generates a graphic user interface (GUI) for controlling the generator tool and plug-in objects, including a database schema access query object and test objects for allowing the GUI to operate with vendor-specific databases. The GUI is "customized" by the various plug-in objects. For example, the database schema access query object is designed specifically for a particular underlying database and retrieves "metadata" concerning the database schema. The retrieved metadata is then displayed as part of the graphic user interface to assist the user in constructing a query object. Test objects are also generated by the GUI in response to a user request. The test objects contain information that characterizes the query object for testing purposes. The information in the test objects is used with a test framework to install and initialize the query object. The test framework also uses the information in the test objects to customize part of the GUI in order to allow a user to view and interact with the installed query object. In particular, the customized GUI allows a developer to enter input parameters for a query directly from the interface and use the installed query object to perform a query with the input parameters. Results which are returned from the query are displayed on the interface.

25 Claims, 15 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 15

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

NAME	Draw Data	Image
------	-----------	-------

☐ 5. Document ID: US 6425121 B1

L2: Entry 5 of 25

File: USPT

Jul 23, 2002

US-PAT-NO: 6425121

DOCUMENT-IDENTIFIER: US 6425121 B1

TITLE: Method and apparatus for resolving divergent paths in graphical programming environments

DATE-ISSUED: July 23, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Phillips; Christopher J.	Walpole	MA		

US-CL-CURRENT: 717/109; 345/762, 345/763, 717/113, 717/133, 717/143, 717/157

ABSTRACT:

The invention relates to a mechanism for detecting forks within data flow diagrams corresponding to application programs and for controlling the execution of such application programs. The flow diagram is preferably formed by interconnecting a plurality of symbolic representations of program objects with a plurality of wire

constructs. The symbolic representations correspond to program objects having input properties and configured to execute pre-defined functions based on the values of its input properties. The flow diagram preferably represents a logical flow of information used to set the input property values of the program objects. According to the invention, the input properties of the program objects are first invalidated. Next, information is allowed to flow along at least a portion of the diagram, thereby setting one or more of the program objects' input property values. In response to having its input property value set, the program object validates the respective input property. The program objects are further configured to wait until all of their input properties have been validated before executing their associated functions.

15 Claims, 25 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 25

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

NAME	Draw Desc	Image
------	-----------	-------

☐ 6. Document ID: US 6425120 B1

L2: Entry 6 of 25

File: USPT

Jul 23, 2002

US-PAT-NO: 6425120
DOCUMENT-IDENTIFIER: US 6425120 B1

TITLE: Repeating program object for use with a graphical program-development system

DATE-ISSUED: July 23, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Morganelli; Michael F.	Norton	MA		
Phillips; Christopher J.	Walpole	MA		
Reilly; Gerard M.	West Roxbury	MA		

US-CL-CURRENT: 717/109; 345/762, 345/763, 717/113, 717/133, 717/143, 717/157

ABSTRACT:

A program object is provided for use in developing application programs through a program-development environment. Using the program-development environment, a developer graphically specifies a flow diagram that represents the logical operation of the application program. The program object, which may have a corresponding symbolic representation for display within the flow diagram, is configured to perform repeating functions and to issue a plurality of ready events, and is preferably used to define loop procedures within the application program. An event handler procedure that is responsive to at least one of the ready events of the repeating program object may be specified by the developer through either graphical inputs or textual inputs to a computer. This event-handler procedure may then be incorporated into the application program for execution at application run-time.

16 Claims, 25 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 25

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

NAME	Draw Desc	Image
------	-----------	-------

☐ 7. Document ID: US 6408430 B2

L2: Entry 7 of 25

File: USPT

Jun 18, 2002

US-PAT-NO: 6408430
DOCUMENT-IDENTIFIER: US 6408430 B2

TITLE: Interactive software testing system and method

DATE-ISSUED: June 18, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Gunter; Elsa L.	Philadelphia	PA		
Peled; Doron A.	Gillette	NJ		

US-CL-CURRENT: 717/109; 717/113

ABSTRACT:

A method for visualizing and testing a sequence of a software code that includes instructions relating to assignment of variables, and decision branches. The method comprises the steps of generating a plurality of nodes, edges and text indications that correspond to the instructions in the software code. A plurality of pointers are then generated to associate the location of at least one line of the software code to at least one of the generated nodes. The pointers may also associate the location of a line in the software code to at least one of the generated edges. The method then displays a flow chart representing the generated nodes, and edges and text indications, so that the software code can be visualized. When a user selects a node or an edge in the displayed flow chart, a corresponding portion of the software code is also identified. In accordance with another embodiment, a portion of the software code defining an execution path is selected, and the system then calculates the logically most general condition for the possible execution of the selected path. In accordance with other embodiments, the path may be selected by a user, who highlights the nodes and/or edges along the displayed flow chart to calculate the conditions that allow the execution of the selected path, or by the system and the results of condition calculations are then presented to the user.

21 Claims, 5 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 5

Full Title Citation Front Review Classification Date Reference Sequences Attachments

NAME Draw Desc Image

☐ 8. Document ID: US 6282699 B1

L2: Entry 8 of 25

File: USPT

Aug 28, 2001

US-PAT-NO: 6282699
DOCUMENT-IDENTIFIER: US 6282699 B1
**** See image for Certificate of Correction ****

TITLE: Code node for a graphical programming system which invokes execution of textual code

DATE-ISSUED: August 28, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Zhang; Roger	Austin	TX		
Dye; Robert	Austin	TX		
McKaskle; Greg	Austin	TX		
Cifra; Chris	Austin	TX		

ABSTRACT:

A system and method for creating a graphical program, wherein the graphical program is operable to invoke execution of textual code. The user selects a code node for inclusion in the graphical program, wherein the code node is displayed on the screen. The user then selects or enters textual code that is comprised in or displayed in the code node. The textual code may be code from a text-based language, such as Perl, Mathematica, or Java, etc., or may be a script from a scripting language. The user may manually enter the textual code into the code node, or import the textual code from a file. The textual code comprised in the code node is user viewable and editable. During execution of the graphical program, the code node is operable to invoke execution of the textual code comprised in the code node. The textual code is preferably executed by an instance of a server program. During execution of the graphical program, the graphical programming system provides the textual code to the server program, as well as any input data received by the code node. The server program then executes the textual code, using any input data received by the code node, and produces an output which is provided back to the graphical program. The present invention thus enables a user of a graphical programming system to more easily incorporate, view, edit and debug textual based code from within the graphical programming system.

28 Claims, 16 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 15

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

WMC	Draw Desc	Image
-----	-----------	-------

☐ 9. Document ID: US 6246404 B1

L2: Entry 9 of 25

File: USPT

Jun 12, 2001

US-PAT-NO: 6246404

DOCUMENT-IDENTIFIER: US 6246404 B1

**** See image for Certificate of Correction ****

TITLE: Automatically generating code for integrating context-sensitive help functions into a computer software application

DATE-ISSUED: June 12, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Feigner; Randall James	Kirkland	WA		
Walden; Ralph Everett	Redmond	WA		

US-CL-CURRENT: 345/708; 717/109, 717/113

ABSTRACT:

A computer software application includes multiple software components such as graphical control objects. A help editor receives context-sensitive help information associated with the graphical control objects. Using this help information as well as information about the graphical control objects, the help editor creates a header file and a map data structure. A help developer edits the header file and map data structure to produce a configuration of context-sensitive help functions for the computer software application. The help editor generates a source code header file and a source code map data structure according to the user-indicated configuration for integrating context-sensitive help functions into the computer software application.

23 Claims, 9 Drawing figures

Exemplary Claim Number: 1

Full Title Citation Front Review Classification Date Reference Sequences Attachments

RMK Draw Desc Image

☐ 10. Document ID: US 6234689 B1

L2: Entry 10 of 25

File: USPT

May 22, 2001

US-PAT-NO: 6234689

DOCUMENT-IDENTIFIER: US 6234689 B1

TITLE: Apparatus and method for mapping a custom routine to an interface button

DATE-ISSUED: May 22, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Rohrbaugh; John G.	Ft. Collins	CO		
Baker; Thomas H.	Ft. Collins	CO		
Bennett; Michael J.	Timnath	CO		
Gil; Mercedes E.	Ft. Collins	CO		
Proulx; Robert W.	Ft. Collins	CO		

US-CL-CURRENT: 717/162; 709/310, 717/113

ABSTRACT:

The present invention is a method for accessing a user defined custom routine through a graphical interface of an application program. The method comprises the steps of: (a) linking the user defined custom routine to the application program; (b) displaying a means for accessing the user defined custom routine on a graphical interface; and (c) transferring control to the user defined custom routine when a user activates the means.

10 Claims, 48 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 48

Full Title Citation Front Review Classification Date Reference Sequences Attachments

RMK Draw Desc Image

☐ 11. Document ID: US 6102967 A

L2: Entry 11 of 25

File: USPT

Aug 15, 2000

US-PAT-NO: 6102967

DOCUMENT-IDENTIFIER: US 6102967 A

TITLE: Testing a help system of a computer software application without executing the computer software application

DATE-ISSUED: August 15, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Feigner; Randall James	Kirkland	WA		
Walden; Ralph Everett	Redmond	WA		

ABSTRACT:

A help editor tests context-sensitive help for graphical controls of dialog boxes of a computer software application without executing the computer software application. The help editor receives and manages a file containing context-sensitive help information text for particular graphical controls. The help editor creates a map of graphical control identifiers to associated help information identifiers. A temporary help file includes the map as well as the context-sensitive help information text. The help editor displays a dialog box of a computer software application. When a user selects a graphical control of the displayed dialog box, the help editor receives a help request. In response, the help editor passes to an operating system help display system a reference to the temporary help file. The operating system help display system causes to be displayed context-sensitive help information for the selected graphical control.

26 Claims, 9 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 8

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

HTML	Draw Desc	Image
------	-----------	-------

☐ 12. Document ID: US 5966532 A

L2: Entry 12 of 25

File: USPT

Oct 12, 1999

US-PAT-NO: 5966532

DOCUMENT-IDENTIFIER: US 5966532 A

TITLE: Graphical code generation wizard for automatically creating graphical programs

DATE-ISSUED: October 12, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
McDonald; Ryan O.	Austin	TX		
Kudukoli; Ramprasad	Austin	TX		
Richardson; Gregory C.	Round Rock	TX		

US-CL-CURRENT: 717/105; 707/101, 717/113

ABSTRACT:

A computer-implemented system and method for automatically generating graphical code in a graphical programming system. The computer memory stores a plurality of graphical code templates. The graphical programming system executing on the computer system also includes a plurality of front panel objects or controls which represent the user interface. One or more associated graphical code portions or templates can be associated with certain of the controls. According to the present invention, the user first selects a control and then preferably initiates the graphical code generation wizard for the control. When the graphical code generation wizard is invoked, the wizard displays on the screen a configuration panel or dialog, prompting the user to configure the control or object. The user then selects parameter values to configure certain aspects of the graphical code being created. The graphical code generation wizard selects a graphical code template in response to the control and configures the graphical code template with the parameter values. The graphical code generation wizard then creates an association between the control and the configured graphical code. The user can edit wizard created code either using the graphical code generation wizard or by unlocking the association between the control and the code and making the changes directly in the block diagram. The present invention also comprises a graphical code generation wizard designed

specifically for industrial automation applications, referred to as the MMI G Wizard.

50 Claims, 34 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 29

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

WWC	Draw Desc	Image
-----	-----------	-------

☐ 13. Document ID: US 5950000 A

L2: Entry 13 of 25

File: USPT

Sep 7, 1999

US-PAT-NO: 5950000
DOCUMENT-IDENTIFIER: US 5950000 A

TITLE: Integrated of a third party software tool

DATE-ISSUED: September 7, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
O'Leary; Daniel J.	Mountain View	CA		
Nelson-Gal; David A.	San Francisco	CA		

US-CL-CURRENT: 717/105; 717/113

ABSTRACT:

Methods, systems, and computer program products for integrating third party tools of an integrated software environment with an integrated environment manager. The graphical user interface of the integrated environment manager is augmented with an icon representing a selected third-party tool. Pulldown and pullright menus are controlled to show objects and files which have been used recently and by which tools. The tools and the integrated environment manager communicate through inter-process messaging to coordinate control of picklists in menus of the integrated environment manager and the tools integrated therewith.

25 Claims, 10 Drawing figures
Exemplary Claim Number: 9
Number of Drawing Sheets: 10

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

WWC	Draw Desc	Image
-----	-----------	-------

☐ 14. Document ID: US 5911072 A

L2: Entry 14 of 25

File: USPT

Jun 8, 1999

US-PAT-NO: 5911072
DOCUMENT-IDENTIFIER: US 5911072 A

TITLE: Method and system for reducing an intentional program tree represented by high-level computational constructs

DATE-ISSUED: June 8, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Simonyi; Charles	Medina	WA		

US-CL-CURRENT: 717/105; 717/113

ABSTRACT:

A method and system for generating a computer program in the manner that uses no computer programming language syntax. The system represents a computer program as an intentional program tree, which is a high-level program tree that is a syntax-independent representation using high-level computational constructs. The intentional program tree represents a programmer's intent, rather than an implementation of the programmer's intent. The programmer creates an intentional program tree using a syntax-independent editor. The editors allows a programmer to directly manipulate the intentional program tree. Because the program is stored as an intentional program tree in a syntax-independent manner, the editor allows the program to select in which of a various programming language the computer program is to be displayed. In addition, the system transforms an intentional program tree to a reduced program tree, which is a program tree comprising low-level computational constructs, in a process called reduction. The reduction process replaces expressions of programmer's intents with a representation of one of possible multiple implementations of those intents using low-level computational constructs.

28 Claims, 38 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 35

Full Title Citation Front Review Classification Date Reference Sequences Attachments

RWMC Draw Desc Image

☐ 15. Document ID: US 5881288 A

L2: Entry 15 of 25

File: USPT

Mar 9, 1999

US-PAT-NO: 5881288
DOCUMENT-IDENTIFIER: US 5881288 A

TITLE: Debugging information generation system

DATE-ISSUED: March 9, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Sumi; Fumio	Katano			JP
Takayama; Shuichi	Takarazuka			JP
Sayama; Junko	Kusatsu			JP
Iwamura; Yoshiyuki	Hirakata			JP
Nagata; Shoji	Hirakata			JP
Nishibata; Motohide	Kadoma			JP

US-CL-CURRENT: 717/125; 717/113

ABSTRACT:

A program development system in which the debugging apparatus is informed of all of the optimization processes which have been performed. A primitive storage unit stores record information for the optimization processes. The input unit receives an input of a variable and a value, or an input of a line where execution is to be halted. The primitive combining unit obtains record information showing the optimization processes. The code execution unit executes the execution code. The variable operation unit obtains the value of a variable based on relations between variables and resources. The output unit displays the obtained value of the variable. The line display unit displays the program or the generated execution code. The line information display unit displays, in line units, information relating to the optimization performed for each line, The operation-possible

variable display unit displays, for each line, variables which can be set and referred to in the line. The optimization process display unit shows how the optimization affects each variable.

24 Claims, 40 Drawing figures
Exemplary Claim Number: 20
Number of Drawing Sheets: 27

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

RWC	Draw Desc	Image
-----	-----------	-------

☐ 16. Document ID: US 5826086 A

L2: Entry 16 of 25

File: USPT

Oct 20, 1998

US-PAT-NO: 5826086
DOCUMENT-IDENTIFIER: US 5826086 A

TITLE: Device and method for aiding designing process of software development

DATE-ISSUED: October 20, 1998

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Arima; Yasuhiko	Sapporo			JP
Nishiyama; Yoshio	Kawasaki			JP
Fukao; Itaru	Kawasaki			JP
Abe; Hiroaki	Kawasaki			JP
Kubota; Yuji	Sapporo			JP

US-CL-CURRENT: 717/105; 345/967, 717/113

ABSTRACT:

A device for aiding a software designing process in which software is divided into a plurality of tasks includes a dynamic-specification-information editing unit for defining dynamic behaviors between the tasks to create a dynamic specification, a static-specification-information editing unit for defining static configurations between the tasks to create a static specification, and an editing-unit-coordination controlling unit for coordinating operations of the dynamic-specification-information editing unit and the static-specification-information editing unit.

17 Claims, 22 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 22

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

RWC	Draw Desc	Image
-----	-----------	-------

☐ 17. Document ID: US 5818711 A

L2: Entry 17 of 25

File: USPT

Oct 6, 1998

US-PAT-NO: 5818711
DOCUMENT-IDENTIFIER: US 5818711 A

TITLE: Method for visually determining the status of program edits in an on-line programming environment

DATE-ISSUED: October 6, 1998

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Schwabe; Judith E.	Cleveland Heights	OH		
Urdaneta; Shelly L.	Lyndhurst	OH		

US-CL-CURRENT: 700/18; 345/866, 700/12, 700/15, 700/86, 715/512, 717/113, 717/173

ABSTRACT:

A method for visually displaying the status of edits utilizes a plurality of visual indicia proximal the portions of edited program segments. Program segments which have not been edited and are currently part of the executable program in the programmable logic controller's processor are represented with solid power rails. Edited rungs which have either been inserted, replaced or deleted are noted as such by I, R and D indicia proximal the power rail of the rung. An asterisk symbol (*) is used to visually identify rungs that have been modified at the workstation but which have not yet been downloaded to the controller. Other visual indicia identify to a workstation operator rungs which have been verified. When the controller is put into test edits mode, the state of the program in the controller prior to downloading the edited program segments is maintained but is not executed and the edits are resolved into the program executing in the processor. Inserted program segments are visually represented as a solid line indicating they are currently part of the program being executed in the controller. Replaced and deleted program segments are visually represented in dotted lines as they are no longer part of the program being executed in the remote processor. In the event the operator deletes a program segment which has been downloaded to the controller, the program segment is visually represented with small d indicia which informs the operator that rung will be deleted at the next download.

14 Claims, 24 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 21

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

FNOC	Dram Desc	Image
------	-----------	-------

☐ 18. Document ID: US 5790863 A

L2: Entry 18 of 25

File: USPT

Aug 4, 1998

US-PAT-NO: 5790863

DOCUMENT-IDENTIFIER: US 5790863 A

TITLE: Method and system for generating and displaying a computer program

DATE-ISSUED: August 4, 1998

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Simonyi; Charles	Medina	WA		

US-CL-CURRENT: 717/113; 700/83, 700/87, 717/114, 717/156

ABSTRACT:

A method and system for generating a computer program. In a preferred embodiment, the present invention provides a program tree editor for directly manipulating a program tree. A program tree comprises of plurality of nodes corresponding to computational constructs. The program tree editor receives commands from a user that are independent of a programming language syntax. The present invention also provides a display representation generator for generating a display representation of the program tree. The display representation generator retrieves nodes from the

program tree and displays a display representation of the node. A user of the present invention preferably interacts with the program tree editor based on the display representation.

15 Claims, 20 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 18

Full Title Citation Front Review Classification Date Reference Sequences Attachments

KWIC Draw Desc Image

☐ 19. Document ID: US 5715432 A

L2: Entry 19 of 25

File: USPT

Feb 3, 1998

US-PAT-NO: 5715432
DOCUMENT-IDENTIFIER: US 5715432 A

TITLE: Method and system for developing network analysis and modeling with graphical objects

DATE-ISSUED: February 3, 1998

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Xu; Jiyang	Superior	CO		
Cox, Jr.; Louis A.	Denver	CO		
Epstein; Michael L.	Boulder	CO		

US-CL-CURRENT: 345/764; 345/853, 345/866, 345/967, 717/109, 717/113, 717/140

ABSTRACT:

A method for developing a network analysis and modeling application program includes the step of interactively selecting and displaying a graphical object. An instantiation of the graphical object is representative of at least a portion of a network. The graphical object is further used in an executable network analysis and modeling application program. The preferred embodiment further includes the step of editing the selection of the graphical object.

6 Claims, 3 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 2

Full Title Citation Front Review Classification Date Reference Sequences Attachments

KWIC Draw Desc Image

☐ 20. Document ID: US 5481712 A

L2: Entry 20 of 25

File: USPT

Jan 2, 1996

US-PAT-NO: 5481712
DOCUMENT-IDENTIFIER: US 5481712 A

TITLE: Method and apparatus for interactively generating a computer program for machine vision analysis of an object

DATE-ISSUED: January 2, 1996

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Silver; William M.	Medfield	MA		
Druker; Samuel	Brookline	MA		
Romanik; Philip	West Haven	CT		
Arbogast; Carroll	Needham	MA		

US-CL-CURRENT: 717/109; 345/810, 717/112, 717/113

ABSTRACT:

A system for interactively generating a computer program for machine vision analysis insures that the program is correct by permitting the operator to make only syntactically correct modifications to the program. The system includes an element for storing the computer program being generated. A further element displays the program to the operator. A positioning element demarks a location of interest within the program. A menu element displays permissible programming modifications for the location of interest. The menu element incorporates in its display of permissible programming modifications statements for machine vision analysis of an object image, e.g., calls to machine vision subroutines and functions. To facilitate specification of input parameters to those subroutines and functions, the imaging element can generate a candidate image of the object upon which the machine vision analysis is to be run. A graphical input element displays over that candidate image a graphical icon that the operator can manipulate to specify the parameters. A textual input element can display an icon, e.g., a dialog box, prompting the operator to designate textually input parameters for the machine vision tool. An update element responds to the operator selection by appropriately modifying the stored program.

34 Claims, 3 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 3

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

FNAC	Draw Desc	Image
------	-----------	-------

☐ 21. Document ID: US 5450545 A

L2: Entry 21 of 25

File: USPT

Sep 12, 1995

US-PAT-NO: 5450545

DOCUMENT-IDENTIFIER: US 5450545 A

TITLE: Generation of rules-based computer programs using data entry screens

DATE-ISSUED: September 12, 1995

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Martin; Robin V.	Portsmouth			GB
Moulton; Michael G.	Portsmouth			GB

US-CL-CURRENT: 717/109; 717/113, 717/117, 717/139, 717/140

ABSTRACT:

A method and system for generating a rules-based computer program employs an interactive workstation to display input data format screens. These screens are edited on the workstation to define the formats of all permissible items of input data to the program and the formats are stored. The workstation then displays rule entry screens in decision table format on which the program designer defines the rules relating to the processing of associated input data items. The completed decision tables are stored and together with the stored input data formats are compiled to produce computer program code for run-time execution or interpretation.

18 Claims, 20 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 19

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

NAME	Draw Desc	Image
------	-----------	-------

☐ 22. Document ID: US 5430873 A

L2: Entry 22 of 25

File: USPT

Jul 4, 1995

US-PAT-NO: 5430873
DOCUMENT-IDENTIFIER: US 5430873 A
**** See image for Certificate of Correction ****

TITLE: Software design support apparatus

DATE-ISSUED: July 4, 1995

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Abe; Hiroaki	Sapporo			JP
Fukao; Itaru	Sapporo			JP
Taneda; Harumi	Kawasaki			JP
Kubota; Yuji	Sapporo			JP
Arima; Yasuhiko	Sapporo			JP
Nakagawa; Naoshi	Sapporo			JP
Konno; Takeo	Sapporo			JP
Arihara; Yoshinori	Sapporo			JP
Suzuki; Yuriko	Sapporo			JP

US-CL-CURRENT: 717/113; 707/102

ABSTRACT:

During a software designing operation, a designer accesses a level prescribing unit through an interaction managing control unit. The level prescribing unit provides the guidance to specification information and design parts, etc. at a desired design level. If the designer selects and inputs specification information or design parts, the design editor corresponding to the design specification information in a design editor unit is activated, so that the designer is guided and aided through a display to produce a desired software. At this time, the information about the state of the editing operation "completed" or "not completed" is stored with the directory. The designed document satisfying a specific condition, for example, a document determined to be "design completed", is stored as a data base by a design data base storing unit. To update the information stored in the design data base, a design information inconsistency correction support unit presents the designer with the part of the information affected by the update, thereby supporting the correction.

36 Claims, 50 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 50

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

NAME	Draw Desc	Image
------	-----------	-------

☐ 23. Document ID: US 5418957 A

L2: Entry 23 of 25

File: USPT

May 23, 1995

US-PAT-NO: 5418957
DOCUMENT-IDENTIFIER: US 5418957 A

TITLE: Network data dictionary

DATE-ISSUED: May 23, 1995

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Narayan; Rom	Wayland	MA	01778	

US-CL-CURRENT: 717/113; 707/1, 717/121

ABSTRACT:

The Network Data Dictionary is a device for enabling standardization of data structures in programs, file layouts and Data Base Management System (DBMS) schema residing in Include Files located on one or more computers in a network. By making the data structures comply with the data element definitions stored in a common data element dictionary, improvements in the quality, accuracy, and consistency of data can be obtained, while simultaneously providing productivity advantages to programmers. The device is set up to organize a set of disparate Include Files (representing data structure descriptions corresponding to program structures, file layouts, and DBMS schema), in one or more computers in a network under a common scheme called the Include File Dictionary, so that these Include Files are made accessible by the device to programmers for sharing, controlled modification, and use. The Include Files are enabled by the device so that programmers can edit these with reference to a common data element dictionary (DED) residing on one of the network nodes. Measurement of the extent to which Include Files correspond to dictionary standards are reported on compilation reports so that corrective actions (such as reconciling conflicting data element definitions) can be taken.

13 Claims, 5 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 5

Full Title Citation Front Review Classification Date Reference Sequences Attachments

RWC Draw Desc Image

☐ 24. Document ID: US 5379430 A

L2: Entry 24 of 25

File: USPT

Jan 3, 1995

US-PAT-NO: 5379430
DOCUMENT-IDENTIFIER: US 5379430 A

TITLE: Object-oriented system locator system

DATE-ISSUED: January 3, 1995

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Nguyen; Frank T.	Campbell	CA		

US-CL-CURRENT: 707/3; 713/2, 717/111, 717/113, 717/120

ABSTRACT:

A method and system for adding system components (documents, tools, fonts, libraries, etc.) to a computer system without running an installation program. A location framework is employed to locate system components whose properties match those specified in a search criteria. The framework receives notification from the system when system components whose properties match the search criteria are added to or removed from the system.

23 Claims, 11 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 11

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

RAWC	Draw Desc	Image
------	-----------	-------

☐ 25. Document ID: US 5327568 A

L2: Entry 25 of 25

File: USPT

Jul 5, 1994

US-PAT-NO: 5327568
DOCUMENT-IDENTIFIER: US 5327568 A

TITLE: Apparatus for supporting graphic data driven program development and for displaying instruction execution results superimposed on the graphic program

DATE-ISSUED: July 5, 1994

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Maejima; Yukihiro	Hachioji			JP
Suzuki; Taihei	Kodaira			JP
Kaneko; Yasuyoshi	Yokohama			JP
Masui; Mitsuyuki	Yokosuka			JP
Kawaguchi; Susumu	Yokohama			JP
Nakatani; Hikaru	Kamakura			JP

US-CL-CURRENT: 717/147; 345/441, 717/113, 717/141

ABSTRACT:

An apparatus for supporting development of a graphic data driven program includes a data driven mechanism enabling instructions of the data driven program to be executed whenever all input data necessary for executing the instructions is available. The apparatus includes a terminal device for inputting and displaying a graphic data driven program, an input transforming section for transforming the inputted graphic data driven program into an intermediate file by referring to an instruction information data base, a wiring table transforming section for transforming the inputted graphic data driven program into a wiring table of instructions expressed in text which is independent of a target machine by referring to the intermediate file, a compile section for transforming the wiring table into an instruction template described with a language depending on the target machine, an instruction execution section for executing the instruction template, an instruction execution displaying section for displaying results of executing the instruction template superposed on the displayed graphic data driven program, and an instruction information tracing section for comparing actual results of executing the instruction template stored in a traced information file with correct results of executing the instruction template stored in a correct data base.

15 Claims, 20 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 14

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

RAWC	Draw Desc	Image
------	-----------	-------

Generate Collection

Print

Terms	Documents
L1 AND testing	25

Display Format:

[Previous Page](#) [Next Page](#)

WEST

[Generate Collection](#)[Print](#)

Search Results - Record(s) 1 through 3 of 3 returned.

☒ 1. Document ID: US 6564368 B1

L4: Entry 1 of 3

File: USPT

May 13, 2003

US-PAT-NO: 6564368

DOCUMENT-IDENTIFIER: US 6564368 B1

TITLE: System and method for visual application development without programming

DATE-ISSUED: May 13, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Beckett; Stephen	Canton	GA		
Lamberti; Santino	Marietta	GA		
Palmer; Todd	Kennesaw	GA		
Beckett; Becky	Canton	GA		
Yoder; Henry	Woodstock	GA		
Donnelly; Ray	Acworth	GA		
McDougald; Danny	Acworth	GA		
House; Donald	Acworth	GA		

US-CL-CURRENT: 717/113; 717/102, 717/104, 717/105, 717/109, 717/110, 717/115, 717/116, 717/125

ABSTRACT:

A method and system for creating distributed applications without programming is disclosed. A Connection Editor interacts with an interface manager attached to each program in the visual development system. The interface manager allows connections to be made using the Connection Editor between the compatible properties of disparate programs by a visual process. Upon connection, the interface manager automates data flow between disparate properties without requiring any additional programming. Furthermore, the interface manager allows for dynamic properties to be exposed based on the run-time discovery of data.

35 Claims, 28 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 28

[Full](#) [Title](#) [Citation](#) [Front](#) [Review](#) [Classification](#) [Data](#) [Reference](#) [Sequences](#) [Attachments](#)

[BIOC](#) [Draw Desc](#) [Image](#)

☐ 2. Document ID: US 6550057 B1

L4: Entry 2 of 3

File: USPT

Apr 15, 2003

US-PAT-NO: 6550057

DOCUMENT-IDENTIFIER: US 6550057 B1

TITLE: Piecemeal retrieval in an information services patterns environment

DATE-ISSUED: April 15, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bowman-Amuah; Michel K.	Colorado Springs	CO		

US-CL-CURRENT: 717/126; 700/80, 707/5, 717/101, 717/102, 717/108, 717/109, 717/113

ABSTRACT:

A system, method and article of manufacture are provided for providing a warning upon retrieval of objects that are incomplete. An object is provided with at least one missing attribute. Upon receipt of a request from an application for the object access to the attributes of the object is allowed by the application. A warning is provided upon an attempt to access the attribute of the object that is missing.

15 Claims, 195 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 123

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

WWW	Draw Desc	Image
-----	-----------	-------

☒ 3. Document ID: US 6505343 B1

L4: Entry 3 of 3

File: USPT

Jan 7, 2003

US-PAT-NO: 6505343

DOCUMENT-IDENTIFIER: US 6505343 B1

TITLE: Document/view application development architecture applied to ActiveX technology for web based application delivery

DATE-ISSUED: January 7, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Menon; Satish R.	Portland	OR		
Muray; Andrew J.	Portland	OR		

US-CL-CURRENT: 717/116; 345/681, 709/201, 709/203, 717/100, 717/101, 717/102, 717/109, 717/113

ABSTRACT:

A method for creating a large functional software component. The method includes the steps of building a large functional unit in a data/data presentation format and wrapping the large functional unit with an encapsulating program, creating a wrapped large functional unit. The method further includes the step of modifying the wrapped large functional unit to become operable with the encapsulating program.

8 Claims, 13 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 9

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

WWW	Draw Desc	Image
-----	-----------	-------

Generate Collection

Print

Terms	Documents
L3 AND L1	3

Display Format: REV Change Format

[Previous Page](#) [Next Page](#)